Iteration 2

Reddit Robot

Alex Harper

Contents

1 Use Cases

Has the previous use cases with minor edits and some extras.

1.1 Authenticate the User (session)

- No Refresh Token
  - Prompt the user to visit the correct URL (includes scopes and program ID)
  - Prompt the user to click "Allow" on the webpage
  - Prompt the user to copy the URL they got redirected to back into the program
  - Parse the URL for the temperary token
  - Use the one-time-use token to get the session and refresh tokens
  - Save the refresh token to disk
  - Restart the session timeout timer
- Has Refresh Token but not Session Token
  - Use the refresh token to get a new session token
  - Restart the session timeout timer
- Has Session Token and Refresh Token
  - If session timeout timer is past 3600 seconds, use *Has Refresh Token but not Session Token*

1.2 Read Data (session)

- *Authenticate the User*
- Check that not too many requests have been made in the last time period
- Set the HTTP header
  - Set the useragent string in the header
  - Set the authorization string in the header
- Make request

- Wait for reply to finish
- Get information from reply
  - Get the *X-Ratelimit-Remaining* number from the header (requests available left)
  - Get the *X-Ratelimit-Reset* number from the header (time until get more requests)
- Return the body of the message

1.3 Getting Subreddit Data

- Ask the session to retrieve data
  - url = "https://oauth.reddit.com/r/" + subreddit + "/about"
  - Use the GET method
- Make sure it is a JSON Object
- Make JSON object and begin the tedium of parsing it into the public variables

1.4 Getting Subreddit Posts

- Ask the session to retrieve data
  - url = "https://oauth.reddit.com/r/" + subreddit
  - Use the GET method
- Make sure it is a JSON Object
- Make JSON object and begin the tedium of parsing it into a list
  - Drill down the JSON stack to get the object for the array of posts
  - Iterate through the array
  - Make a new Post object for every index
  - Add the new Post to the list
- Return the list of Posts

1.5 Getting Post Comments

- Ask the session to retrieve data
  - url = "https://oauth.reddit.com/" + permalink
  - Use the GET method
- Make sure it is a JSON Object
- Make JSON object and begin the tedium of parsing it into a list
  - Drill down the JSON stack to get the object for the array of comments
  - Iterate through the array
  - Make a new Comment object for every index
  - Add the new Comment to the list
- Return the list of Comments

1.6 Getting Current Account Data

- Ask the session to retrieve data
  - url = "https://oauth.reddit.com/r/api/v1/me"
  - Use the GET method
- Make sure it is a JSON Object
- Make JSON object and begin the tedium of parsing it into the public variables

1.7 Other Cases I want to add later

- Getting a Certain Account's data
- Imgur read and write with their API
- Message content to parse for in my bot

2 Domain Diagram

Figure 1: The Original Reddit Domain Diagram. It is mostly correct, but for accounts it is missing details.

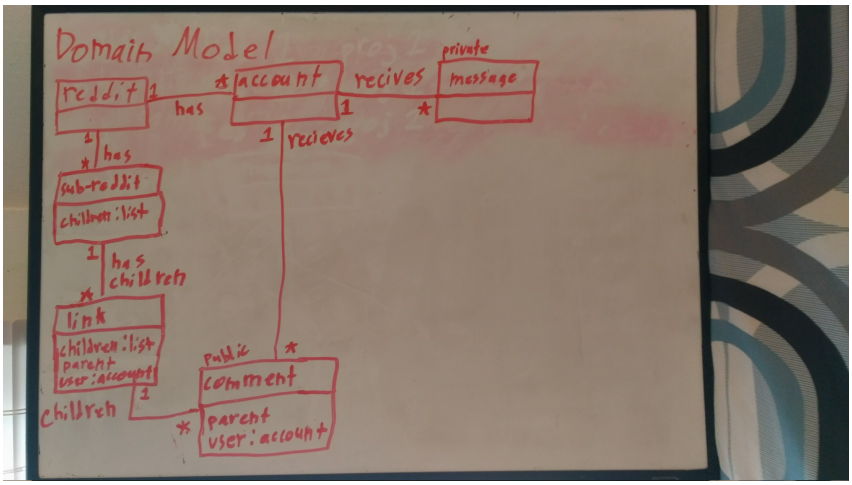
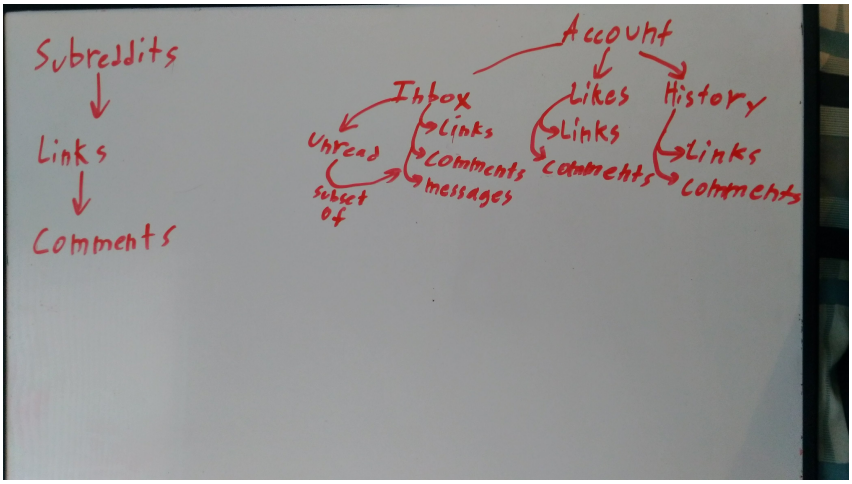


Figure 2: The New Reddit Domain Diagram. This was made in the middle of the iteration when I had figured out all the parts.



3 Reading Data

Figure 3: Original Sequence Diagram Example for Getting Data. It was at a time when JSON was not really thought about, and simply that strings of some sort would come back.

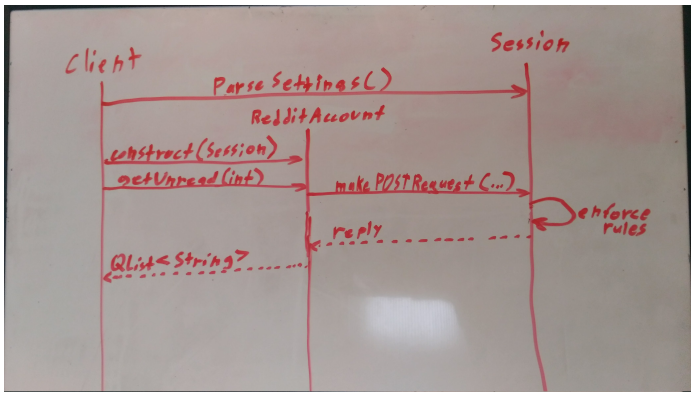
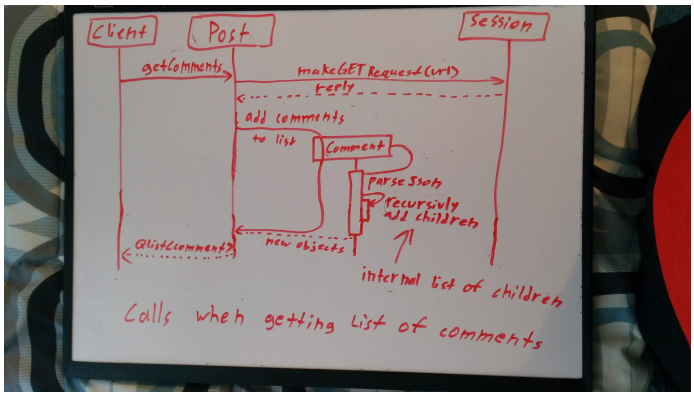


Figure 4: New Sequence Diagram Example for Getting Data. It has the most interesting example where the comments recursivly parse for new things. Many other calls are just simpler versions of this.



4 Object Structure

Figure 5: Thinking about Object Responsibilities. I made this to just keep my thoughts straight in the middle of coding one day.

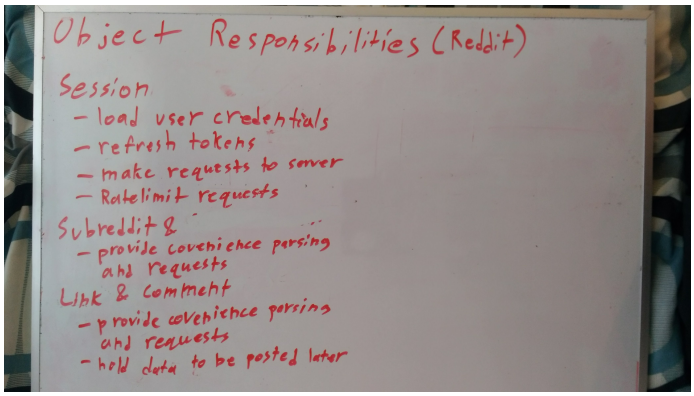
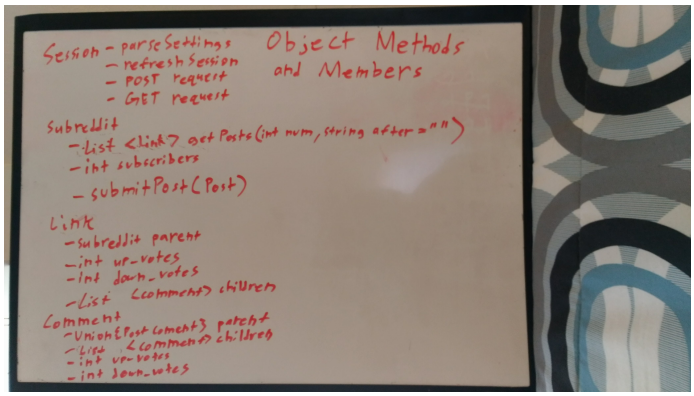


Figure 6: Implementing Responsibilities. I made this a little later that day on my second board. It was me thinking about some basics to implement the responsibilities.



5 JSON

Figure 7: Original Skectch for JSON Parsing. You can see me using (wrong) notation to show it as a base class of the other objects.

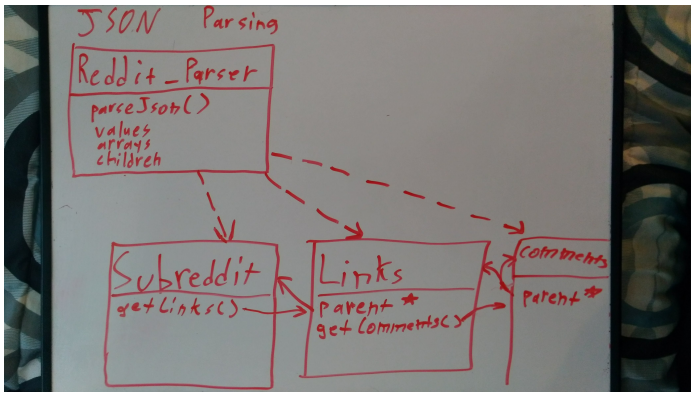
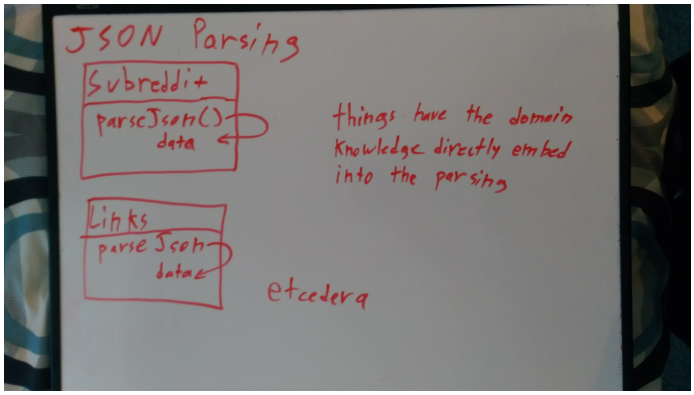


Figure 8: JSON Parsing Now. I drew this for this report just as a comparison to the old thing. Objects just do it independently and it is proably less code than if I pre-parsed.



6 Client Example Code

Figure 9: Idealized API Code Usage. I like to do this sometimes for different things. It makes me do a sanity check for what I want the API of the objects I'm making to be. I guess this is ensuring proper responsibilities if I were to talk in terms of the book, but historically I just felt that things were weird if I meander through coding.